

Modern Engineering

March 06 | Day 8

Prudential Financial



Whilst we wait....

In chat write:

Your name - role and a recent impulse buy.

**Let's give everyone
a min or two to join**



Data Flow and Routing in React

LESSON ROADMAP



Unidirectional
Data Flow

Fruits Filter
Code-along

React
Routing

Routing Lab

MEF MODULE 3 DAY 8: Forms and Routing in React

Schedule	
9:00–9:15 am	Welcome and Warm-Up
9:15 am–12:30 pm	Filtering Fruits Application
12:30–1:30 pm	Lunch
1:30–2:30 pm	React Router
2:30–3:30 pm	Pokedex Routing
3:30 - 4:50 pm	React Fundamentals Review Lab
4:50–5:00 pm	Bring It Home

LEARNING OBJECTIVES

1

Identify which components should contain state and how to pass it down to children components

2

Implement the "lifting state" pattern allowing children components to impact ancestor's state

3

Use react router to allow url-based navigation within a single page application

4

Apply specific route parameters to allow dynamic, id-based paths in a react application



Routing and Forms in React



Unidirectional Data Flow





[Let's head to github](#) to tackle the complexities of React's unidirectional data flow and discover the pattern commonly known as "lifting state" between components.



Lunch Time

Modern Engineering



Routing



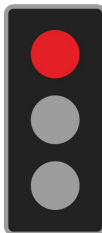
Single Page Applications and the User Experience

While the reduction in server load is a nice benefit of Single Page Applications, arguably the greatest benefit is the enhanced user experience.

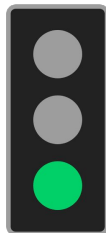
Avoiding full-page reloads creates a dramatically smoother user experience, particularly after the first initial page load. From there, the app just relies on JavaScript and JSON data from a backend server to create a dynamic, responsive web application.



MapQuest vs. Google Maps



- Navigating on a map required clicking a button to go North, South, East or West
- The entire page would reload with the new coordinates!
- Sloooooow experience



- Google Maps created a SPA experience that loaded the new map with JavaScript instead
- No reloading, easy navigation

Traditional Web Apps

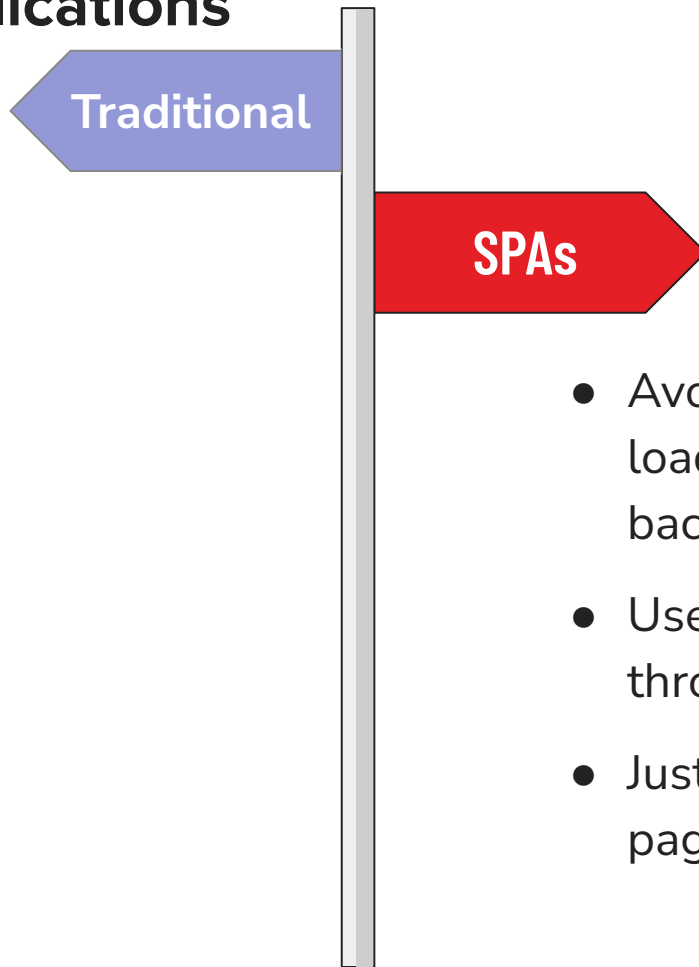


Traditional

SPAs

- Each new page sends a new request to the server
- Pages can be easily shared and bookmarked
- Each page has its own url and HTML file

Single Page Applications



- Avoids repeated page loads, only sends JSON back and forth
- Users navigate to a page through actions
- Just one initial URL for the page

Fake It Till You Make It

Being able to link to a specific "page" within a single-page application is a vital part of the user experience...but there's still only one page!

Solution: simulate the old way of doing things by having the application become responsive to the url provided in the address bar. Users will never know the difference!



Routing

React Router is a popular library for routing in React applications.

- It provides a way to define and manage multiple routes within a single-page application.
- This allows the user to navigate between different views or pages of the application.

Question: Based on the example provide, where does this route take users?

```
function App() {  
  return (  
    <Router>  
      <div>  
        <nav>  
          <ul>  
            <li>  
              <Link to="/">Home</Link>  
            </li>  
            <li>  
              <Link to="/about">About</Link>  
            </li>  
            <li>  
              <Link to="/users">Users</Link>  
            </li>  
          </ul>  
        </nav>  
  
        <Route path="/" exact component={Home} />  
        <Route path="/about" component={About} />  
        <Route path="/users" component={Users} />  
      </div>  
    </Router>  
  )  
}
```

Image: Code block of React Router



Guided Walk-Through: Routing a PokeDex

[Let's head to github](#) for a walkthrough on using Router on a SPA that pulls in pokemon data from an API.



Solo Exercise:

Routing in Your App

60 minutes



Most CRUD applications have the same standard user-facing routes. For example, a **/pokemon/new** route in a user-facing application would lead to a form that lets users create a new pokemon.

Research the RESTful routes that correspond to your resources and implement as many of them as possible in your React application.





Discussion: React Review

2 hours



We now have all the tools we're going to use to build our full-stack React + Express application!

Let's recap by walking back through some of the apps we've built and solidifying any conceptual questions remaining before our big build tomorrow.





Solo Exercise:

Daily Exit Ticket

3 minutes



- Please take a moment to give us your feedback after today's course!
- Use the QR code or follow the link: <https://bit.ly/pruMEF-c3>



Scan Me!

